Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 223 (2009) 672–692
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

A Newton-type method for constrained least-squares data-fitting with easy-to-control rational curves

G. Casciola[a], L. Romani[b,*]

[a] *Department of Mathematics, University of Bologna, P.zza di Porta San Donato 5, 40127 Bologna, Italy*
[b] *Department of Mathematics and Applications, University of Milano-Bicocca, Via R. Cozzi 53, 20125 Milano, Italy*

Received 7 May 2007; received in revised form 11 December 2007

Abstract

While the mathematics of constrained least-squares data-fitting is neat and clear, implementing a rapid and fully automatic fitter that is able to generate a fair curve approximating the shape described by an ordered sequence of distinct data subject to certain interpolation requirements, is far more difficult.

The novel idea presented in this paper allows us to solve this problem with efficient performance by exploiting a class of very flexible and easy-to-control piecewise rational Hermite interpolants that make it possible to identify the desired solution with only a few computations. The key step of the fitting procedure is represented by a fast Newton-type algorithm which enables us to automatically compute the weights required by each rational piece to model the shape that best fits the given data. Numerical examples illustrating the effectiveness and efficiency of the new method are presented.

MSC: 65D10; 65D17; 41A20; 41A29; 90C51; 90C30

Keywords: Least-squares data-fitting; Constrained approximation; Rational Hermite interpolation; Automatic selection of weights; Interior-point Newton-type method

1. Introduction

In several applications of computer-aided geometric design we frequently have to deal with 3D point reconstruction problems where a large amount of data is given in the form of an ordered sequence of distinct points describing a target shape in space. Most of these usually contain measurement errors, while only a few are rigorously generated and then turn out to be crucial to the final reconstruction.

As regards the computation of a surface/surface intersection curve, for example, the set of discrete data consists of highly-accurate points (such as border points, turning points and cusp points) precisely detected on the intersection curve, and a sequence of marching points generated from each of the above points by going a step in the direction defined by the local differential geometry of the curve. The first ones, commonly called starting points, will be therefore exactly interpolated, while all the others will be approximated in order to reduce to a minimum the sum of the squares of their distances from the desired curve.

* Corresponding author.
E-mail addresses: casciola@dm.unibo.it (G. Casciola), lucia.romani@unimib.it (L. Romani).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2008.02.005

An analogous situation occurs in the point-based construction of a Gordon-type surface. This time the input data is given by a set of 3D points, describing the complete cross-sections of the target surface, which have to be accurately approximated to produce a fair curve network to be assumed as the surface skeleton. Consistency demands that the cross-section curves agree in value where an x-section crosses a y-section. This means that sequences of points along two transversal directions must possess common intersections and these have to be assumed as positional constraints for the fitting problem.

In both the outlined circumstances – as well as in many other applications – it is therefore desirable to use a fitting method that, on one hand, is able to capture the shape of the overall input data and, on the other, to satisfy the assigned point constraints precisely.

Since, when approximating shapes with a complicated behaviour, it always turns out to be convenient to construct the fitting curve segment-wise by means of a piecewise model defined in the space of conventional polynomial splines or NURBS, our idea consists of using the point constraints (identified by the specific application we are considering, or detected from the curvature and torsion information of the input dataset) to partition the given data into adjacent subsets that can be approximated separately by a curve segment taken from some specified class of appropriate curves.

Taking into account that when the points lie on the intersection of two analytic surfaces, derivative constraints of any order can be easily computed by their parametric representation and, when generating Gordon-type surfaces, additional information like first derivatives and/or higher order derivatives to be assumed by the curve network in the significant locations are also generally available, the most natural solution to this kind of constrained fitting problem can be obtained by using a fitter that implements a piecewise Hermite interpolant. This solution also allows us to greatly simplify the computational process that a standard least-squares minimization problem with associated positional constraints would have required in order to ensure a sufficiently high order of continuity at segment boundaries.

While for pure interpolation there is probably little reason to use a rational form, for approximation purposes allowing weights to be arbitrary makes it possible to produce fitting curves with higher accuracy and fewer control points.

The novel solution we are going to propose will rely therefore on a class of piecewise rational Hermite interpolants. In particular we will adopt here the one introduced in [1,2] because of its flexibility and ease of control. This can also be represented in the conventional NURBS form by assuming multiple knots in the correspondence of the location of the interpolation constraints and letting the control points be dependent both on them and on the weights of the rational representation. In this way, once a procedure for computing the optimal weight values has been designed, the control points turn out to be automatically defined and hence the best-fitting curve results completely determined.

Therefore, differently from standard NURBS fitting procedures, which require a complicated and expensive iterative algorithm to minimize (with respect to knots, control points and weights) a sum of squared Euclidean norms measuring the distance between the point set and the curve to be generated [3,4,9,11–15], the least-squares fitting method we are going to propose will be performed exclusively to identify the choice of weights that guarantees the best reconstruction of the original data. Moreover, while the output of existing algorithms cannot always guarantee a fitting curve with a fair shape (namely with a curvature plot consisting of only a small number of monotone pieces), due to the definition of the novel fitter this follows easily and, whenever the degree of the curve primitive is larger than three, a curvature-continuous approximation of the original data is also ensured.

The organization of the paper is as follows. In Section 2 we introduce the rational Hermite basis to be used as a novel curve primitive for determining the solution of the constrained least-squares problem. Next in Section 3 we develop a strategy for carrying out the automatic computation of the optimal weights to be embodied in the desired rational form, and we describe the overall fitting process in all its steps. Finally, in Section 4 we close the paper by showing some numerical examples that confirm the improved performance of the innovative procedure relative to conventional and reliable approaches (like the well-known `lsqcurvefit` algorithm that is currently implemented in MATLAB's Optimization Toolbox).

2. Least-squares fitting with a novel curve primitive

Given a set of 3D distinct points representing a target shape in space, we seek a NURBS curve that lies close to the assigned data and passes through only a few of them. Let $\{\mathbf{Q}_k\}_{k=0,\dots,M-1}$ denote the given set of points in $\mathbb{R}^3$ and $\Im \subset \{0,\dots,M-1\}$ the subset of $N+1$ ($N \ll M$) indexes specifying all the points $\mathbf{Q}_k$ that the fitting curve must

interpolate. In the following we will use the notation $\{\mathbf{F}_i\}_{i=0,\dots,N}$ to identify the point constraints $\mathbf{Q}_\Im$ and we will also assume that the first and last data points are always interpolated, that is $\mathbf{Q}_0 \equiv \mathbf{F}_0$ and $\mathbf{Q}_{M-1} \equiv \mathbf{F}_N$. Hence the point set $\{\mathbf{Q}_k\}_{k=0,\dots,M-1}$ turns out to be partitioned by $\mathbf{Q}_\Im$ into N adjacent subsets that can be approximated separately by a curve segment $\mathbf{c}_i(t)$ taken from some specified class of admissible curves. Since it is required that segments $\mathbf{c}_i$ have common endpoints $\mathbf{F}_i$, $\mathbf{F}_{i+1}$ and a contact of order ℓ at $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$, the C^ℓ-continuous piecewise curve $\mathbf{c}(t) = \bigcup_{i=0}^{N-1} \mathbf{c}_i(t)$ can be naturally generated through a piecewise rational Hermite interpolant of degree $n = 2\ell + 1$. As, in practice, a reconstruction with piecewise curves of low degree is usually preferred, due to their simplicity and robustness, we will confine ourselves here to consider rational Hermite models defined by $n = 3$ and $n = 5$ only. This choice is also supported by the observation that such degrees allow us to establish a good compromise between the number of curve pieces $\mathbf{c}_i(t)$ and the accuracy of the final fitting $\mathbf{c}(t)$. By taking into account that, if a large number of pieces is used a curve with very small fitting errors is obtained, and vice versa if the number of pieces is too small the fitting errors might be very large, it is clear that the number of curve segments should be controlled in such a way that the fitting errors reach a level that the user can accept. Since the idea behind the algorithm we are going to propose is to fit a degree-n rational Hermite segment to the data confined between the interpolating points $\mathbf{F}_i$, $\mathbf{F}_{i+1}$, and, if this cannot guarantee the desired level of accuracy, to adaptively subdivide the point set limited by the assigned constraints and reconsider the subsets (repeating the procedure until the required error tolerance holds), it is important to make a choice of n which can guarantee a successful reconstruction that does not need too many and computationally expensive segments. To this aim it has been proved experimentally that, if n is chosen equal to 3 or 5, a tight and economical fit is always ensured through only a restricted number of curve pieces. From now on we will therefore address our attention towards the cubic and the quintic piecewise rational Hermite models only. Note that the same choice was also made in [7], where piecewise Hermite polynomials were originally adopted as fitting curve basis as an alternative to classical B-splines. But in this work, to facilitate local approximation with endpoint constraints, we will let each single Hermite piece to be represented in the rational Bézier form. Next, in order to get a standard NURBS representation, degree-n Bézier segments will be pieced together with n-fold knots in correspondence of data points $\{\mathbf{F}_i\}_{i=0,\dots,N}$ where the joins take place [1,2].

As frequently done with data-fitting procedures, the overall knot-partition $\{\tau_k\}_{k=0,\dots,M-1}$ is obtained by applying the cumulative chord length parameterization

$$\tau_k = \tau_{k-1} + \frac{\|\mathbf{Q}_k - \mathbf{Q}_{k-1}\|_2}{\sum_{j=1}^{M-1} \|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|_2} = \frac{\sum_{j=1}^{k} \|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|_2}{\sum_{j=1}^{M-1} \|\mathbf{Q}_j - \mathbf{Q}_{j-1}\|_2}, \quad k = 1, \dots, M-1 \tag{1}$$

(with $\tau_0 = 0$) to the given point set $\mathbf{Q}_k$, and the subset of multiple break knots $\{t_i\}_{i=0,\dots,N}$, used to identify the junctions between consecutive rational pieces, is computed by selecting from the previous one the location parameters of index $\Im$. By construction it results therefore that $t_0 < t_1 < \cdots < t_N$ and $t_0 \equiv \tau_0$, $t_N \equiv \tau_{M-1}$.

Any Hermite model always requires that ℓth-order derivatives $\mathbf{D}_i^{(\ell)}$ ($\ell \geq 1$) are assigned in correspondence of the interpolating points $\{\mathbf{F}_i\}_{i=0,\dots,N}$; whenever they are not given as constraints or cannot be easily obtained (as happens when the given data lie on the intersection of two analytic surfaces), then a data-sensitive derivative estimation must be computed as part of the fitting algorithm. While estimating first derivatives has been a subject of extensive study, an appropriate estimation of higher order derivatives is still considered a difficult task. But as it was assumed that $n \in \{3, 5\}$ and neighboring Bézier segments are joined with a level of continuity related to the degree of the fitting curve through the formula $\ell = \frac{n-1}{2}$, it turns out to be sufficient to estimate the first and second order derivatives only. To this aim we are allowed to exploit a 3-point strategy, based on local quadratic interpolation, that enables us to derive them compatibly with the behaviour of the data. In particular, given the three-dimensional point set $\mathbf{Q}_k$ and the sequence of location parameters τ_k, an appropriate estimation of the first and second order derivatives $\{\mathbf{D}_i^{(\ell)}\}_{i=0,\dots N}$ ($\ell = 1, 2$) at points $\mathbf{F}_i$, can be worked out respectively by firstly computing the direction vectors

$$\Delta^{(1)}\mathbf{Q}_k = \frac{\theta_k\ \Delta\mathbf{Q}_{k-1} + \theta_{k-1}\ \Delta\mathbf{Q}_k}{\theta_{k-1} + \theta_k} \tag{2}$$

and

$$\Delta^{(2)}\mathbf{Q}_k = \frac{2(\Delta\mathbf{Q}_k - \Delta\mathbf{Q}_{k-1})}{\theta_{k-1}+\theta_k}, \tag{3}$$

(where $\theta_k = \tau_{k+1} - \tau_k$ and $\Delta\mathbf{Q}_k = \frac{\mathbf{Q}_{k+1}-\mathbf{Q}_k}{\theta_k}$ for all $k = 0, \ldots, M-1$) and successively selecting from them the $N+1$ values defined in correspondence of the knot subsequence $\{t_i\}_{i=0,\ldots,N}$.

Remark 1. In the above strategy ℓth-order derivatives at τ_k are estimated to be those of the quadratic which passes through $(\tau_j, \mathbf{Q}_j)$, $j = k-1, k, k+1$. Thus, in the case of open curves, the auxiliary point required by Eqs. (2) and (3) at both the endpoints can be consistently derived by a quadratic extrapolation rule. In the case of closed curves, we simply assume $\mathbf{Q}_{-1} = \mathbf{Q}_{M-2}$ and $\mathbf{Q}_M = \mathbf{Q}_1$.

At this point, having identified the subset of break knots $\{t_i\}_{i=0,\ldots,N}$ defined in correspondence of the significant points $\{\mathbf{F}_i\}_{i=0,\ldots,N}$ and computed the associated derivatives $\{\mathbf{D}_i^{(\ell)}\}_{i=0,\ldots,N}$ (for all $\ell = 1, \ldots, \frac{n-1}{2}$), we aim to construct N rational Hermite curve segments $\{\mathbf{c}_i(t)\}_{i=0,\ldots,N-1}$ of a chosen degree n ($n \in \{3, 5\}$) that interpolate the assigned end constraints and best approximate the ordered sequence of points lying in between. To this purpose it turns out to be very helpful to express each single Hermite piece $\mathbf{c}_i(t) : [t_i, t_{i+1}] \to \mathbb{R}^3$ into the following rational Bézier form.

Definition 2. Defined the overall knot-partition $\{\tau_k\}_{k=0,\ldots,M-1}$ and the related subsequence of break knots $\{t_i\}_{i=0,\ldots,N}$, the degree-n ($n \in \{3, 5\}$) rational Hermite interpolant $\mathbf{c}_i(t)$, with $t \in [t_i, t_{i+1}]$, can be conveniently written in the rational Bézier form

$$\mathbf{c}_i(t) = \sum_{j=0}^{n} \mathbf{P}_j^i R_{j,n}^i(t), \tag{4}$$

where $\{\mathbf{P}_j^i\}_{j=0,\ldots,n}$ denote the $n+1$ control points associated with the basis functions

$$R_{j,n}^i(t) = \frac{w_j^i B_{j,n}^i(t)}{\sum_{h=0}^{n} w_h^i B_{h,n}^i(t)} \tag{5}$$

defined via the positive weights

$$w_0^i = 1, \qquad \left\{w_j^i\right\}_{j=1,\ldots,n-1}, \qquad w_n^i = 1, \tag{6}$$

and the degree-n Bernstein polynomials

$$B_{j,n}^i(t) = \binom{n}{j} \frac{(t_{i+1}-t)^{n-j}\,(t-t_i)^j}{(\eta_i)^n}, \quad \text{with } \eta_i = t_{i+1} - t_i. \tag{7}$$

Since, once the degree n is fixed, an explicit formulation of all Bézier control points $\{\mathbf{P}_j^i\}_{j=0,\ldots,n}$ in terms of the weights $\{w_j^i\}_{j=1,\ldots,n-1}$ and of the assigned boundary constraints $\left(t_i, \mathbf{F}_i, \{\mathbf{D}_i^{(\ell)}\}_{\ell=1,\ldots,\frac{n-1}{2}}\right)$, $\left(t_{i+1}, \mathbf{F}_{i+1}, \{\mathbf{D}_{i+1}^{(\ell)}\}_{\ell=1,\ldots,\frac{n-1}{2}}\right)$ can be provided, we will now formalize their definition both in the cubic and in the quintic cases.

Definition 3. A degree-3 rational Hermite interpolant $\mathbf{c}_i(t)$ of the kind (4) is defined by control points $\{\mathbf{P}_j^i\}_{j=0,\ldots,3}$ having the following expressions:

$$\mathbf{P}_0^i = \mathbf{F}_i, \qquad \mathbf{P}_1^i = \mathbf{F}_i + \frac{\eta_i \mathbf{D}_i^{(1)}}{3w_1^i}, \qquad \mathbf{P}_2^i = \mathbf{F}_{i+1} - \frac{\eta_i \mathbf{D}_{i+1}^{(1)}}{3w_2^i}, \qquad \mathbf{P}_3^i = \mathbf{F}_{i+1}. \tag{8}$$

Analogously,

Definition 4. A degree-5 rational Hermite interpolant $\mathbf{c}_i(t)$ of the kind (4) is defined by control points $\{\mathbf{P}^i_j\}_{j=0,\dots,5}$ having the following expressions:

$$\mathbf{P}^i_0 = \mathbf{F}_i, \qquad \mathbf{P}^i_1 = \mathbf{F}_i + \frac{\eta_i \mathbf{D}^{(1)}_i}{5w^i_1}, \qquad \mathbf{P}^i_2 = \mathbf{F}_i + \frac{(5w^i_1 - 1)\eta_i \mathbf{D}^{(1)}_i}{10w^i_2} + \frac{\eta^2_i \mathbf{D}^{(2)}_i}{20w^i_2},$$

$$\mathbf{P}^i_3 = \mathbf{F}_{i+1} - \frac{(5w^i_4 - 1)\eta_i \mathbf{D}^{(1)}_{i+1}}{10w^i_3} + \frac{\eta^2_i \mathbf{D}^{(2)}_{i+1}}{20w^i_3}, \qquad \mathbf{P}^i_4 = \mathbf{F}_{i+1} - \frac{\eta_i \mathbf{D}^{(1)}_{i+1}}{5w^i_4}, \qquad \mathbf{P}^i_5 = \mathbf{F}_{i+1}. \tag{9}$$

As regards the rational cubic primitive, it follows by definition that it gives C^1 smoothness, which is sufficient for many applications, while keeping computational costs to a minimum. However, as for certain applications higher order curves could be necessary or advantageous (in general because higher smoothness and better approximation are required), whenever C^2 continuity is needed the rational quintic model will be able to provide the desired solution. Despite each piece of the quintic being slightly more expensive than its degree-3 correspondent, it has excellent approximation properties and makes it possible to reduce the number of pieces needed to fit a target shape to the same tolerance.

In the next sections we will show how the proposed rational models can be advantageously used to construct an optimal fitting of 3D data points, such that high accuracy is guaranteed and a compact representation is pursued.

3. A Newton-type optimization algorithm for computing best-fitting weights

We now introduce the notation $\{\mathbf{Q}^i_k\}_{k=0,\dots,M_i-1}$ to identify the M_i points $\mathbf{Q}_k$ confined between two assigned interpolating points $\mathbf{F}_i, \mathbf{F}_{i+1}$, and we denote by $\{\tau^i_k\}_{k=0,\dots,M_i-1}$ the parameters τ_k associated with these points (such that $\mathbf{Q}^i_0 = \mathbf{F}_i$, $\mathbf{Q}^i_{M_i-1} = \mathbf{F}_{i+1}$ and $\tau^i_0 = t_i$, $\tau^i_{M_i-1} = t_{i+1}$). In this way, the approximating curve $\mathbf{c}(t) = \bigcup_{i=0}^{N-1} \mathbf{c}_i(t)$ will be made of single pieces $\mathbf{c}_i(t)$ in the form (4), defined to minimize over each interval $[t_i, t_{i+1}]$ the least-squares error

$$\Phi(w^i_1, \dots, w^i_{n-1}) = \sum_{k=0}^{M_i-1} \|\mathbf{c}_i(\tau^i_k) - \mathbf{Q}^i_k\|^2_2 \tag{10}$$

with respect to the unknowns $\{w^i_j\}_{j=1,\dots,n-1}$. Since the dependence of the ith segment $\mathbf{c}_i(t)$ on the weight vector $\mathbf{w}^i = (w^i_1, \dots, w^i_{n-1})^{\mathrm{T}}$ of the rational representation (4) is nonlinear and is not available in a simple analytical form, we have to use a nonlinear optimization procedure that can cope with this problem. But as the range of positive values attainable by the parameters w^i_j should be bounded away from zero and, for clear practical reasons, limited by a plausible upper bound, indeed we have to deal with a minimization problem whose solutions can only be expected in a particular area. We are thus allowed to consider a numerical method for nonlinear optimization with a feasible set in the form of a box. However, differently from standard NURBS curves, we are not forced to avoid using dramatically varying weights, since, being the control points definition influenced by the weights themselves, whatever their choice is we can guarantee a good parameterization. In this way we can arbitrarily set lower and upper bounds, giving the weights the possibility of taking values very distant from 1. To this purpose it has been proved experimentally that a good choice of a feasible set is given by the box

$$\Omega := \{\mathbf{w}^i \in \mathbb{R}^{n-1} : \ l_j \le w^i_j \le u_j \ \text{ with } \ l_j = 10^{-3}, \ u_j = 10^3 \ \forall j = 1, \dots, n-1\},$$

where l_j, u_j denote the lower and upper bounds for w^i_j, respectively. While $l_j = 10^{-3}$ has been imposed by the condition that weights w^i_j should be strictly positive and sufficiently far from zero, the value $u_j = 10^3$ is dictated by a reasonable choice that allows us to simplify our analysis and make the solution useful in practical problems. Indeed we have to solve a nonlinear and multivariate optimization problem that aims at minimizing the objective function $\Phi : \mathbb{R}^{n-1} \to \mathbb{R}$, defined by the following infinitely continuously differentiable expression

$$\Phi(\mathbf{w}^i) = \sum_{k=0}^{M_i-1} \|\mathbf{c}_i(\tau^i_k) - \mathbf{Q}^i_k\|^2_2 = \sum_{k=0}^{M_i-1} \|\mathbf{E}^i_k\|^2_2, \tag{11}$$

subject to $\mathbf{w}^i \in \Omega$.

The existence of local minima is generally not a real difficulty in our case, but a fast convergence towards a minimum is certainly a crucial problem.

Usually, most efficient algorithms for calculating a local minimum of a nonlinear function may be considered to be *descent methods*. At each step they consist in minimizing the objective function along a straight line defined by a direction named the *descent-direction*. Each method is characterized by the way in which this direction is built. Be warned that, if it is not opportunely defined, the algorithm may become really expensive and time-consuming since several iterations may be needed to reach the minimum.

When the objective function is very regular, among the fastest descent methods we can find the so-called *Newton-type methods*. In fact, when the exact computation of the objective function derivatives is possible, strategies which do not use this information turn out to be more expensive than the ones which do.

In principle, for NURBS curve fitting problems, computing derivatives with respect to weights poses no severe difficulties, primarily because NURBS are a rational combination of these variables. In addition, using the special NURBS representation proposed in the previous section, the gradient and Hessian of the objective function turn out to possess a very simple and compact symbolic form. Thus, to perform the optimization process in (11) it appears to be convenient to use a Newton-type algorithm that requires the computation of both first and second order derivatives of $\Phi(\mathbf{w}^i)$. Indeed, due to the presence of box constraints, the most adequate class of Newton-type algorithms turns out to be the one of *projected affine-scaling interior-point Newton methods*.

In the last ten years several papers proposing more and more efficient variants of this type of simple-constrained optimization procedure have been published (see [8,10] and references therein).

Among all possible solvers of this class, we consider here the iterative method introduced in [8] and we develop some improvements to guarantee that the desired fitting curve is obtained with less computation and a fast convergence is ensured in any situation. In particular, we firstly observe that there is a scaling matrix that can be cancelled on both sides of the linear system to be solved at each step of Heinkenschloss et al. algorithm, allowing a more efficient implementation; secondly, since when choosing a value of the steplength σ very close to 1 (as suggested by the authors) their iterative method does not always converge to the optimal weight vector $\mathbf{w}^{i,*}$, we will propose an innovative optimization procedure that can combine the above-mentioned simplified version of the algorithm in [8] with a modified choice of the σ parameter that is able to ensure an order-two convergence for any type of input data. Our solution to this kind of problem is based on a non-stationary definition of σ which, although starting from a small guess in (0, 1), allows us to guarantee in only a few steps increasing values closer and closer to 1, thus providing an affine-scaling interior-point Newton-type method that is always quadratically convergent.

Before developing the novel order-two interior-point Newton method that enables us to compute the best-fitting values of $\mathbf{w}^i$, we will introduce the following notation. For the objective function $\Phi : \mathbb{R}^{n-1} \to \mathbb{R}$, we denote by $\nabla\Phi(\mathbf{w}^i) \in \mathbb{R}^{n-1}$ and $\nabla^2\Phi(\mathbf{w}^i) \in \mathbb{R}^{(n-1)\times(n-1)}$ its gradient vector and its Hessian matrix, respectively, while we use $\left[\nabla\Phi(\mathbf{w}^i)\right]_j$ and $\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{j_1 j_2}$ for their components.

By definition, the gradient of $\Phi(\mathbf{w}^i)$ is the $(n-1)$ dimensional column vector

$$\nabla\Phi(\mathbf{w}^i) = \begin{pmatrix} \dfrac{\partial\Phi(\mathbf{w}^i)}{\partial w_1^i} \\ \vdots \\ \dfrac{\partial\Phi(\mathbf{w}^i)}{\partial w_{n-1}^i} \end{pmatrix}, \tag{12}$$

while its Hessian is the $(n-1)\times(n-1)$ symmetric matrix

$$\nabla^2\Phi(\mathbf{w}^i) = \begin{pmatrix} \dfrac{\partial^2\Phi(\mathbf{w}^i)}{\partial(w_1^i)^2} & \cdots & \dfrac{\partial^2\Phi(\mathbf{w}^i)}{\partial w_1^i \partial w_{n-1}^i} \\ \vdots & \cdots & \vdots \\ \dfrac{\partial^2\Phi(\mathbf{w}^i)}{\partial w_{n-1}^i \partial w_1^i} & \cdots & \dfrac{\partial^2\Phi(\mathbf{w}^i)}{\partial(w_{n-1}^i)^2} \end{pmatrix}. \tag{13}$$

Even though providing an explicit formulation of $\nabla\Phi(\mathbf{w}^i)$ and $\nabla^2\Phi(\mathbf{w}^i)$ for arbitrary objective functions $\Phi(\mathbf{w}^i)$ is generally a difficult and computationally intensive task, one of the beauties of the rational model proposed here for representing $\mathbf{c}_i(t)$, is that it allows us to remarkably simplify these operations. In fact, using Eq. (11) and the sum and product derivative rules, we can state that

$$\nabla\Phi(\mathbf{w}^i) = \begin{pmatrix} 2\sum_{k=0}^{M_i-1} \mathbf{E}_k^i \bullet \dfrac{\partial \mathbf{E}_k^i}{\partial w_1^i} \\ \vdots \\ 2\sum_{k=0}^{M_i-1} \mathbf{E}_k^i \bullet \dfrac{\partial \mathbf{E}_k^i}{\partial w_{n-1}^i} \end{pmatrix}, \tag{14}$$

where $\bullet$ denotes the inner product of two vectors. Then, in turn, using the difference and quotient derivative rules, we can assert that for a rational cubic Hermite element it holds that

$$\frac{\partial \mathbf{E}_k^i}{\partial w_1^i} = -\frac{[\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{1,3}^i(\tau_k^i)}{\sum_{h=0}^{3} w_h^i\, B_{h,3}^i(\tau_k^i)}, \tag{15}$$

$$\frac{\partial \mathbf{E}_k^i}{\partial w_2^i} = -\frac{[\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{2,3}^i(\tau_k^i)}{\sum_{h=0}^{3} w_h^i\, B_{h,3}^i(\tau_k^i)}, \tag{16}$$

while for its quintic correspondent we have

$$\frac{\partial \mathbf{E}_k^i}{\partial w_1^i} = -\frac{[\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{1,5}^i(\tau_k^i) + \frac{\eta_i}{2}\mathbf{D}_i^{(1)}B_{2,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i\, B_{h,5}^i(\tau_k^i)}, \tag{17}$$

$$\frac{\partial \mathbf{E}_k^i}{\partial w_2^i} = -\frac{[\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{2,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i\, B_{h,5}^i(\tau_k^i)}, \tag{18}$$

$$\frac{\partial \mathbf{E}_k^i}{\partial w_3^i} = -\frac{[\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{3,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i\, B_{h,5}^i(\tau_k^i)}, \tag{19}$$

$$\frac{\partial \mathbf{E}_k^i}{\partial w_4^i} = -\frac{[\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{4,5}^i(\tau_k^i) - \frac{\eta_i}{2}\mathbf{D}_{i+1}^{(1)}B_{3,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i\, B_{h,5}^i(\tau_k^i)}. \tag{20}$$

Thus, inserting Eq. (15), (16) and (17)–(20) in (14), we get respectively

$$\nabla\Phi(\mathbf{w}^i) = \begin{pmatrix} -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{1,3}^i(\tau_k^i)}{\sum_{h=0}^{3} w_h^i B_{h,3}^i(\tau_k^i)} \\ -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{2,3}^i(\tau_k^i)}{\sum_{h=0}^{3} w_h^i B_{h,3}^i(\tau_k^i)} \end{pmatrix} \tag{21}$$

in the cubic case, and

$$\nabla\Phi(\mathbf{w}^i) = \begin{pmatrix} -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{1,5}^i(\tau_k^i) + \frac{\eta_i}{2}\mathbf{E}_k^i \bullet \mathbf{D}_i^{(1)} B_{2,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i B_{h,5}^i(\tau_k^i)} \\ -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)]B_{2,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i B_{h,5}^i(\tau_k^i)} \\ -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{3,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i B_{h,5}^i(\tau_k^i)} \\ -2\sum_{k=0}^{M_i-1} \dfrac{\mathbf{E}_k^i \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)]B_{4,5}^i(\tau_k^i) - \frac{\eta_i}{2}\mathbf{E}_k^i \bullet \mathbf{D}_{i+1}^{(1)} B_{3,5}^i(\tau_k^i)}{\sum_{h=0}^{5} w_h^i B_{h,5}^i(\tau_k^i)} \end{pmatrix} \tag{22}$$

in the quintic one. Hence, the elements in the Hessian matrix $\nabla^2\Phi(\mathbf{w}^i)$ can be reduced to the simplified expressions written below. Be warned that, for ease of notation, we will omit the arguments of the Bernstein basis functions $B_{j,n}^i(\tau_k^i)$ $(j = 1, \ldots, n-1)$ and we will assume the following compact forms:

$$\begin{aligned}
&\mathcal{G}_{1,k}^i := [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)] \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i) + 2\mathbf{E}_k^i],\\
&\mathcal{G}_{2,k}^i := [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)] \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i) + 2\mathbf{E}_k^i],\\
&\mathcal{H}_k^i := [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i)] \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i) + \mathbf{E}_k^i] + [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i)] \bullet \mathbf{E}_k^i,\\
&\mathcal{I}_{1,k}^i := \mathbf{D}_i^{(1)} \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i) + \mathbf{E}_k^i],\\
&\mathcal{I}_{2,k}^i := \mathbf{D}_{i+1}^{(1)} \bullet [\mathbf{F}_i - \mathbf{c}_i(\tau_k^i) + \mathbf{E}_k^i],\\
&\mathcal{J}_{1,k}^i := \mathbf{D}_i^{(1)} \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i) + \mathbf{E}_k^i],\\
&\mathcal{J}_{2,k}^i := \mathbf{D}_{i+1}^{(1)} \bullet [\mathbf{F}_{i+1} - \mathbf{c}_i(\tau_k^i) + \mathbf{E}_k^i],\\
&\mathcal{K}_1^i := \mathbf{D}_i^{(1)} \bullet \mathbf{D}_i^{(1)}, \qquad \mathcal{K}_2^i := \mathbf{D}_i^{(1)} \bullet \mathbf{D}_{i+1}^{(1)}, \qquad \mathcal{K}_3^i := \mathbf{D}_{i+1}^{(1)} \bullet \mathbf{D}_{i+1}^{(1)}.
\end{aligned}$$

In this way, in the cubic case it holds that

$$\begin{aligned}
\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{11} &= 2\sum_{k=0}^{M_i-1} \frac{\left(B_{1,3}^i\right)^2 \mathcal{G}_{1,k}^i}{\left(\sum_{h=0}^{3} w_h^i B_{h,3}^i\right)^2},\\
\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{12} &= 2\sum_{k=0}^{M_i-1} \frac{B_{1,3}^i B_{2,3}^i \mathcal{H}_k^i}{\left(\sum_{h=0}^{3} w_h^i B_{h,3}^i\right)^2},\\
\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{22} &= 2\sum_{k=0}^{M_i-1} \frac{\left(B_{2,3}^i\right)^2 \mathcal{G}_{2,k}^i}{\left(\sum_{h=0}^{3} w_h^i B_{h,3}^i\right)^2},
\end{aligned}$$

while in the quintic one we get

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{11} = 2\sum_{k=0}^{M_i-1}\frac{\left(B^i_{1,5}\right)^2\mathcal{G}^i_{1,k}+\eta_i B^i_{1,5}B^i_{2,5}\mathcal{I}^i_{1,k}+\frac{\eta_i^2}{4}\left(B^i_{2,5}\right)^2\mathcal{K}^i_1}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{12} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{1,5}B^i_{2,5}\mathcal{G}^i_{1,k}+\frac{\eta_i}{2}\left(B^i_{2,5}\right)^2\mathcal{I}^i_{1,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{13} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{1,5}B^i_{3,5}\mathcal{H}^i_k+\frac{\eta_i}{2}B^i_{2,5}B^i_{3,5}\mathcal{J}^i_{1,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{14} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{1,5}B^i_{4,5}\mathcal{H}^i_k+\frac{\eta_i}{2}(B^i_{2,5}B^i_{4,5}\mathcal{J}^i_{1,k}-B^i_{1,5}B^i_{3,5}\mathcal{I}^i_{2,k})-\frac{\eta_i^2}{4}B^i_{2,5}B^i_{3,5}\mathcal{K}^i_2}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{22} = 2\sum_{k=0}^{M_i-1}\frac{\left(B^i_{2,5}\right)^2\mathcal{G}^i_{1,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{23} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{2,5}B^i_{3,5}\mathcal{H}^i_k}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{24} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{2,5}B^i_{4,5}\mathcal{H}^i_k-\frac{\eta_i}{2}B^i_{2,5}B^i_{3,5}\mathcal{I}^i_{2,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{33} = 2\sum_{k=0}^{M_i-1}\frac{\left(B^i_{3,5}\right)^2\mathcal{G}^i_{2,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{34} = 2\sum_{k=0}^{M_i-1}\frac{B^i_{3,5}B^i_{4,5}\mathcal{G}^i_{2,k}-\frac{\eta_i}{2}\left(B^i_{3,5}\right)^2\mathcal{J}^i_{2,k}}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2},$$

$$\left[\nabla^2\Phi(\mathbf{w}^i)\right]_{44} = 2\sum_{k=0}^{M_i-1}\frac{\left(B^i_{4,5}\right)^2\mathcal{G}^i_{2,k}-\eta_i B^i_{3,5}B^i_{4,5}\mathcal{J}^i_{2,k}+\frac{\eta_i^2}{4}\left(B^i_{3,5}\right)^2\mathcal{K}^i_3}{\left(\sum_{h=0}^{5}w^i_h B^i_{h,5}\right)^2}.$$

Since the Newton-type algorithm we are going to describe is an iterative method that updates the weight vector $\mathbf{w}^i$ at each step s, until the optimal one $\mathbf{w}^{i,*}$ is determined, we will denote the solution correspondent to the sth step by

$\mathbf{w}^{i,s}$. Afterwards we use the explicit expressions of $\nabla\Phi(\mathbf{w}^i)$ and $\nabla^2\Phi(\mathbf{w}^i)$ developed above to define the following diagonal matrices for the sth round of the algorithm:

- $D(\mathbf{w}^{i,s}) := \operatorname{diag}\left(d_1(w_1^{i,s}), \ldots, d_{n-1}(w_{n-1}^{i,s})\right)$, where for any $j = 1, \ldots, n-1$

$$d_j(w_j^{i,s}) := \begin{cases} \tilde{d}_j(w_j^{i,s}), & \text{if } |[\nabla\Phi(\mathbf{w}^{i,s})]_j| < \min\{w_j^{i,s} - l_j, u_j - w_j^{i,s}\}^2 \\ & \text{or } \min\{w_j^{i,s} - l_j, u_j - w_j^{i,s}\} < |[\nabla\Phi(\mathbf{w}^{i,s})]_j|^2, \\ 1, & \text{otherwise} \end{cases}$$

with

$$\tilde{d}_j(w_j^{i,s}) := \begin{cases} w_j^{i,s} - l_j, & \text{if } [\nabla\Phi(\mathbf{w}^{i,s})]_j > 0, \\ u_j - w_j^{i,s}, & \text{if } [\nabla\Phi(\mathbf{w}^{i,s})]_j < 0, \\ \min\{w_j^{i,s} - l_j, u_j - w_j^{i,s}\}, & \text{if } [\nabla\Phi(\mathbf{w}^{i,s})]_j = 0; \end{cases}$$

- $G(\mathbf{w}^{i,s}) := \operatorname{diag}\left(g_1(w_1^{i,s}), \ldots, g_{n-1}(w_{n-1}^{i,s})\right)$, where for any $j = 1, \ldots, n-1$

$$g_j(w_j^{i,s}) := \begin{cases} |[\nabla\Phi(\mathbf{w}^{i,s})]_j|, & \text{if } |[\nabla\Phi(\mathbf{w}^{i,s})]_j| < \min\{w_j^{i,s} - l_j, u_j - w_j^{i,s}\}^2 \\ & \text{or } \min\{w_j^{i,s} - l_j, u_j - w_j^{i,s}\} < |[\nabla\Phi(\mathbf{w}^{i,s})]_j|^2, \\ 0, & \text{otherwise.} \end{cases}$$

As with any iterative scheme, to start the algorithm good initial guesses are required for the unknowns. Namely, we want to start with a plausible configuration of the variables $w_j^{i,0}$ with respect to the observation that the weight vector must contain values in the range $[l_j, u_j]$. In practice, we have found in our experiments that using the initial vector $\mathbf{w}^{i,0} = (1, \ldots, 1)^{\mathrm{T}}$ is adequate for any cases. Therefore, the iterative process we are going to illustrate always starts from a piecewise polynomial Hermite model – since all the weights w_j^i are initialized to 1 – and proceeds updating their values through iterative minimization of the fitting error.

The other main issue in minimizing an objective function is given, instead, by the stop criterion. To check the convergence towards the target shape, the stop criterion

$$\frac{|\Phi(\mathbf{w}^{i,s}) - \Phi(\mathbf{w}^{i,s-1})|}{\Phi(\mathbf{w}^{i,s})} < \delta \quad \forall i = 0, \ldots, N-1 \tag{23}$$

on the squared 2-norms of the residuals at the current and last step is examined over each segment at the same time as the termination test on the 2-norm of the following scaled gradient

$$\|D(\mathbf{w}^{i,s})\,\nabla\Phi(\mathbf{w}^{i,s})\|_2 < \varepsilon \quad \forall i = 0, \ldots, N-1. \tag{24}$$

The stop occurs when these norms are lower than the given accuracies. In order to see some interesting effects when comparing this method with a competitive one, for the trial we have chosen for tolerances δ and ε the values 10^{-10} and 10^{-25}, respectively. These parameter values may be changed by the user if a higher or lower accuracy is required. Note that, for practical reasons, the second criterion is more meaningful, but the first one gives better information about the quality of the minimum when the function is flat around it.

Having fixed starting values and stopping criterions, we are then in the position to state the novel interior-point Newton-type method for the solution of the bound constrained optimization problem (11).

Algorithm 1 (*Best-fitting weights computation*).

(S.0) Choose tolerances δ and ε (we have set $\delta = 10^{-10}$ and $\varepsilon = 10^{-25}$ in our implementation). Set $s := 0$, $\mathbf{w}^{i,0} := (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^{n-1}$ and $\Phi(\mathbf{w}^{i,-1}) = 2\Phi(\mathbf{w}^{i,0})$. Define a starting parameter $\sigma^0 \in (0, 1)$ (we have found $\sigma^0 = 0.3$ to be a good choice for any kind of data).

(S.1) Compute $\Psi(\mathbf{w}^{i,s}) := \frac{|\Phi(\mathbf{w}^{i,s}) - \Phi(\mathbf{w}^{i,s-1})|}{\Phi(\mathbf{w}^{i,s})}$ and $\mathbf{b}(\mathbf{w}^{i,s}) := D(\mathbf{w}^{i,s})\,\nabla\Phi(\mathbf{w}^{i,s})$. If $\Psi(\mathbf{w}^{i,s}) < \delta$ and $\|\mathbf{b}(\mathbf{w}^{i,s})\|_2 < \varepsilon$, STOP.

(S.2) Compute $A(\mathbf{w}^{i,s}) := D(\mathbf{w}^{i,s})\,\nabla^2\Phi(\mathbf{w}^{i,s}) + G(\mathbf{w}^{i,s})$.

(S.3) Let $\mathbf{x}^s \in \mathbb{R}^{n-1}$ be the Newton-type search direction obtained by solving the linear system $A(\mathbf{w}^{i,s})\,\mathbf{x}^s = \mathbf{b}(\mathbf{w}^{i,s})$.

(S.4) Compute $P_\Omega(\mathbf{w}^{i,s} - \mathbf{x}^s)$ using the following projection mapping:

$$P_\Omega : \mathbb{R}^{n-1} \to \Omega$$
$$P_\Omega(w_j^{i,s} - x_j^s) = \max\left\{l_j, \min\{u_j, w_j^{i,s} - x_j^s\}\right\} \quad \forall j = 1, \ldots, n-1. \tag{25}$$

(S.5) Compute $\rho^s := \max\left\{\sigma^s, 1 - \|P_\Omega(\mathbf{w}^{i,s} - \mathbf{x}^s) - \mathbf{w}^{i,s}\|_2\right\}$, so that strict feasibility of the iterates worked out in the following step can be ensured.

(S.6) Set $\mathbf{w}^{i,s+1} := \mathbf{w}^{i,s} + \rho^s\left(P_\Omega(\mathbf{w}^{i,s} - \mathbf{x}^s) - \mathbf{w}^{i,s}\right)$.

(S.7) Update the value of σ^s through the formula

$$\sigma^{s+1} = \sqrt{\frac{1+\sigma^s}{2}}. \tag{26}$$

(S.8) Set $s \leftarrow s+1$ and go to (S.1).

Remark 5. Note that the linear system to be solved in step (S.3) contains just $n-1$ equations (i.e. 2 and 4 equations in the cubic and quintic case respectively), that is it can be easily solved by a direct approach like Gaussian elimination.

Lemma 6. *Given an initial parameter $\sigma^0 \in (0,1)$, the recurrence relation in (26) satisfies the properties:*

$$\sigma^s \in (0,1) \quad \forall s \geq 1 \tag{27}$$

and

$$\lim_{s\to+\infty} \sigma^s = 1. \tag{28}$$

Proof. Demonstration of (27) trivially follows.

To prove (28) we recall that a monotonic and bounded sequence is always convergent and, in particular, if it is non decreasing and upper bounded, then it converges to the upper bound of the values it assumes. For the recurrence formula

$$\begin{cases} \sigma^0 \in (0,1) \\ \sigma^s = \sqrt{\dfrac{1+\sigma^{s-1}}{2}} \quad \forall s \geq 1 \end{cases} \tag{29}$$

it holds that the sequence $\{\sigma^s\}_{s\geq 1}$ is non decreasing and hence convergent to 1. In fact, called γ its limit, we have

$$\gamma = \lim_{s\to+\infty} \sigma^s = \lim_{s\to+\infty}\left(\sqrt{\frac{1+\sigma^{s-1}}{2}}\right) = \sqrt{\frac{1+\gamma}{2}}.$$

Thus, solving the last equation with respect to γ we get $\gamma = 1$. □

From the above result it follows that $\sigma^s \in (0,1)$ for any step s and the sequence generated in (S.7) converges to 1 so that Heinkenschloss et al. condition on the steplength is still satisfied. Thus the projected direction is truncated by a coefficient ρ^s which approaches fast enough to 1 that the following convergence result holds.

Proposition 7. *The weight vector $\mathbf{w}^{i,s}$ converges with order two to the best-fitting solution $\mathbf{w}^{i,*}$.*

Proof. Since the parameter σ^s defined in (26) always stays in $(0,1)$ and the linear system to be solved in step (S.3) is equivalent to the one derived by Heinkenschloss et al. in [8], the same convergence result established for that procedure still holds for the projected affine-scaling interior-point Newton method proposed in Algorithm 1. □

By combining the σ parameter correction in Lemma 6 with the simplified version of Heinkenschloss et al. algorithm, it is possible to get an optimization method that is very efficient and extremely robust. Due to Proposition 7, the convergence theory from Heinkenschloss et al. paper still holds for this approach.

We now assemble the quadratically-convergent iterative method in Algorithm 1 with the steps described in Section 2 in order to develop a constrained least-squares fitting technique that aims at generating the piecewise rational Hermite curve $\mathbf{c}(t)$ which passes as close as possible to the sequence of points $\mathbf{Q}_k$ assigned as input and interpolates the set of points $\mathbf{F}_i$ and derivatives $\mathbf{D}_i^{(\ell)}$ ($\ell = 1, 2$) defined as constraints.

Algorithm 2 (*Constrained Least-squares Data-Fitting*).

Input:

- the degree n ($n \in \{3, 5\}$) of the desired fitting curve (remember that if C^ℓ continuity is required, then the chosen degree must be $n = 2\ell + 1$);
- a 3D point set $\{\mathbf{Q}_k\}_{k=0,\dots,M-1}$ and a subset of interpolating points $\{\mathbf{F}_i\}_{i=0,\dots,N}$ with $\mathbf{F}_0 \equiv \mathbf{Q}_0$ and $\mathbf{F}_N \equiv \mathbf{Q}_{M-1}$;
- optionally, ℓth-order derivatives $\{\mathbf{D}_i^{(\ell)}\}_{i=0,\dots,N}$ ($\ell = 1, 2$) to be assumed by the fitting curve in correspondence of the interpolating points $\{\mathbf{F}_i\}_{i=0,\dots,N}$;
- a parameter $\lambda = 0, 1, 2$ specifying the order of assigned derivatives (note that $\lambda = 0$ means no derivatives are specified).

1. *Chord length parameterization.*
   Determine the parameters τ_k to be associated with the points $\{\mathbf{Q}_k\}_{k=0,\dots,M-1}$ through the chord-length method in (1) and select from them the subsequence of break knots $\{t_i\}_{i=0,\dots,N}$, fixing $t_0 = \tau_0$ and $t_N = \tau_{M-1}$, such that the endpoints of the data set $\mathbf{Q}$ always match with the endpoints of the fitting curve $\mathbf{c}(t)$.
2. *Check specified constraints.*
   2.1. If $\lambda = 0$
   Compute an appropriate estimation of derivatives $\{\mathbf{D}_i^{(\ell)}\}_{i=0,\dots,N}$, for all $\ell = 1, \dots, \frac{n-1}{2}$, by selecting among the direction vectors $\{\Delta^{(\ell)}\mathbf{Q}_k\}_{k=0,\dots,M-1}$ derived through formulas (2)–(3), the ones corresponding to the subsequence of knots $\{t_i\}_{i=0,\dots,N}$.
   2.2. Elseif $\lambda = 1$ and $n > 3$
   Compute an appropriate estimation of 2nd-order derivatives $\{\mathbf{D}_i^{(2)}\}_{i=0,\dots,N}$ by selecting among the direction vectors $\{\Delta^{(2)}\mathbf{Q}_k\}_{k=0,\dots,M-1}$ derived through formula (3), the ones corresponding to the subsequence of knots $\{t_i\}_{i=0,\dots,N}$.
3. *Best-fitting weights computation.*
   For all $i = 0, \dots, N-1$ approximate the data points $\{\mathbf{Q}_k^i\}_{k=0,\dots,M_i-1}$ lying between two consecutive interpolating points $\mathbf{F}_i, \mathbf{F}_{i+1}$, through the rational hermite interpolant $\mathbf{c}_i(t)$ with optimal parameters $\{w_j^i\}_{j=1,\dots,n-1}$ provided by Algorithm 1.

Output:
the degree-n approximating curve $\mathbf{c}(t) = \bigcup_{i=0}^{N-1} \mathbf{c}_i(t)$ in the desired form (4).

To summarize, the idea at the base of our innovative strategy is the following: (i) we define a number of parameter values compatible with the given data set, (ii) we estimate first and, if required, second order derivatives in correspondence of all points to be precisely interpolated and (iii) finally, for each curve piece, we compute the weight vector which minimizes the fitting error in the least-squares sense, so that the rational Bézier form (4) can be provided. The main contribution of this paper lies therefore in bringing all these components together with some innovations to complete a new and efficient algorithm for constrained least-squares data-fitting. Its performances in terms of accuracy, number of iterations and computing time have been extensively analyzed over a wide range of experiments which confirm its superiority if compared with existing procedures. In the next section we will illustrate the results we obtained by comparing the proposed fitter with MATLAB's `lsqcurvefit` function.

4. Comparisons and experimental results

To illustrate the efficiency of the fitting method derived in Section 3, we have compared the optimal weights computed in Algorithm 1 with MATLAB's `lsqcurvefit` procedure. By default this implements a subspace trust region approach based on the interior-reflective Newton method described in [5,6], whose single step involves the approximate solution of a large linear system by using the method of Preconditioned Conjugate Gradients (PCG).

The two algorithms have been implemented in MATLAB and tested for fitting many sets of points with constraints, always showing that, starting with the same initialization values and adopting the same stopping criterions, the novel approach allows us to reduce the number of `lsqcurvefit` iterations and the overall computing time significantly, while keeping the fitting curve within at least the same accuracy. Similar results, proving the superior performance of our method, have been obtained testing the two procedures on many sequences of points representing any type of target shape in space.

Since important applications in Computer Aided Geometric Design often require fitting points that lie on the intersection of two surfaces by using a fair and accurate NURBS model, in the examples selected for this paper we have exploited degree 3 and 5 rational Hermite elements to approximate sequences of points $\{\mathbf{Q}_k\}_{k=0,\dots,M-1}$ generated by a marching algorithm arising from a set of starting points (consisting of border points, turning points and cusp points) that turns out to be crucial to the intersection curve reconstruction.

For each segment $\mathbf{c}_i(t)$ the closeness of fit has been verified by computing the average distance of the rational curve representation, resulting by the identified weights, from the data set $\mathbf{Q}$. Such a distance has been worked out with respect to the sequence of parameter values τ_k previously determined through (1), by means of the root mean square error

$$E^i_{\mathrm{RMS}} = \left(\frac{1}{M_i}\ \Phi(\mathbf{w}^{i,*})\right)^{\frac{1}{2}}. \tag{30}$$

The efficiency of the new procedure in terms of approximation accuracy, number of iterations and computing time is confirmed by numerical results listed in the tables provided (they all refer to tests made on a Pentium IV 3 GHz PC computer; be warned that as regards the `lsqcurvefit` procedure, the parameter Nit includes also the number of PCG iterations). As was confirmed by this and many other runs, each round of Algorithm 1 turns out to be really effective, so that convergence to the optimum is always achieved in a few steps which overall require a very short time. More precisely, while in the cubic case the same approximation accuracy reached by the `lsqcurvefit` procedure is obtained in slightly less rounds of the algorithm, the overall computing time needed to work out the best-fitting weights, is noticeably smaller. In the quintic case the superiority of the new approach is even more evident: more precise fitting curves are generated through a remarkably reduced number of iterations which leads to a significant improvement of the computational time.

Data points considered in Table 1–2, 3–4, 5–6 and 7 are determined respectively by a free-form surface/free-form surface, a sphere/free-form surface, a sphere/cylinder and a torus/cylinder intersection. In these four cases the overall set $\mathbf{Q}$ is made of 2302, 702, 514 and 590 points with $\mathbf{Q}_0 \equiv \mathbf{Q}_{M-1}$; the subset of starting points assumed in each of these contexts is given by 7, 5, 5 and 7 entries, respectively. For all the data sets the sequence of break points $\mathbf{F}_i$ to be interpolated by the final fitting coincides with the given starting points (see Fig. 1, Figs. 3 and 5 and Tables 1, 3 and 5). Sometimes, however, since the resulting number of curve pieces turns out to be too small to get a sufficiently accurate reconstruction, wherever required we adaptively subdivide the point set $\{\mathbf{Q}^i_k\}_{k=0,\dots,M_i-1}$ confined between the assigned constraints $\mathbf{F}_i$ and $\mathbf{F}_{i+1}$, in such a way that the fitting error of each curve segment reaches the required error tolerance. In the examples provided this strategy has been applied on all the 4 subsets in data set 2 and 3 to make the E^i_{RMS} errors related to the least-squares cubic procedure reach the same level of accuracy obtained by its quintic counterpart (see Fig. 4, Fig. 6 and Tables 4–6). Parallel to this, the first, second, fifth and sixth of the 6 subsets identified by the 7 starting points adopted for data set 1, have been subdivided to generate 10 pieces of cubic with E^i_{RMS} errors smaller than 10^{-2} (see Fig. 2 and Table 2). Analogously, the second and fifth of the 6 subsets identified by the 7 starting points adopted for data set 4, have been subdivided to generate 8 pieces of quintic with E^i_{RMS} errors smaller than 10^{-2} (see Fig. 7 and Table 7). Note that the best fitting curve obtained by running the `lsqcurvefit` algorithm on data set 4 subject to the so computed point constraints, is not able to satisfy the required accuracy.

In all the selected examples first and second order derivatives at the prescribed locations $\mathbf{F}_i$ have been worked out through formulas (2)–(3) as previously explained in Section 2. Since the novel fitter relies on piecewise Hermite curves, interpolation of computed derivatives allows us to naturally guarantee a fair fitting, as proved by the curvature and torsion plots in Figs. 8–11.

The innovative least-squares fitting method proposed in Section 3 is thus optimized for handling arbitrary spatial data sets with constraints in respect of great quality and high speed.

Table 1
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 1 for computing the best-fitting weights of the rational quintic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	462	8.502508e−03	[0.966060, 1.542797, 2.879184, 3.966312]	862	
$i=1$	404	6.623846e−03	[0.651238, 0.443307, 0.371919, 0.304374]	550	
$i=2$	286	1.764809e−03	[2.606749, 2.858480, 1.995475, 0.378754]	133	5.375
$i=3$	223	3.290281e−03	[1.172632, 0.700361, 1.492148, 2.419566]	79	
$i=4$	470	4.656866e−03	[1.743332, 1.381264, 0.960209, 1.059620]	37	
$i=5$	462	5.929960e−03	[2.440559, 1.722728, 0.971568, 0.749022]	94	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	462	8.444008e−03	[1.105368, 1.694575, 3.560147, 5.058892]	13	
$i=1$	404	6.576148e−03	[0.599338, 0.404932, 0.353357, 0.305320]	11	
$i=2$	286	1.747577e−03	[2.690090, 2.948713, 2.049340, 0.380522]	11	1.000
$i=3$	223	2.897061e−03	[0.671066, 0.323021, 0.651753, 1.329987]	8	
$i=4$	470	4.358240e−03	[1.105289, 0.837489, 0.590726, 0.688098]	7	
$i=5$	462	5.911690e−03	[2.538682, 1.774742, 0.972904, 0.753972]	9	

Table 2
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 1 for computing the best-fitting weights of the rational cubic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	203	9.901198e−03	[0.810957, 0.916647]	7	
$i=1$	260	1.655288e−03	[1.057301, 1.079532]	7	
$i=2$	230	6.240072e−03	[0.955543, 0.891114]	7	
$i=3$	175	7.886030e−03	[0.657198, 0.616490]	9	
$i=4$	286	9.376179e−03	[2.794953, 2.642180]	11	0.734
$i=5$	223	7.159967e−03	[0.554349, 0.757921]	9	
$i=6$	240	2.843597e−03	[1.407494, 1.393785]	9	
$i=7$	231	6.872127e−03	[0.792434, 0.815610]	7	
$i=8$	328	6.596095e−03	[1.120769, 1.080506]	7	
$i=9$	135	9.978804e−03	[0.976209, 0.821802]	7	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	203	9.901198e−03	[0.810957, 0.916646]	4	
$i=1$	260	1.655288e−03	[1.057300, 1.079531]	5	
$i=2$	230	6.240072e−03	[0.955543, 0.891114]	4	
$i=3$	175	7.886030e−03	[0.657198, 0.616489]	5	
$i=4$	286	9.376179e−03	[2.794966, 2.642193]	9	0.516
$i=5$	223	7.159967e−03	[0.554346, 0.757920]	5	
$i=6$	240	2.843597e−03	[1.407494, 1.393785]	7	
$i=7$	231	6.872127e−03	[0.792434, 0.815609]	4	
$i=8$	328	6.596095e−03	[1.120768, 1.080505]	5	
$i=9$	135	9.978804e−03	[0.976210, 0.821802]	5	

Table 3
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 2 for computing the best-fitting weights of the rational quintic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	175	9.851175e−03	[0.799363, 0.854799, 0.855022, 0.806935]	388	
$i=1$	177	9.787529e−03	[0.809450, 0.863250, 0.863679, 0.810923]	310	2.656
$i=2$	177	1.140690e−02	[0.786697, 0.840007, 0.837876, 0.789397]	373	
$i=3$	176	1.260546e−02	[0.780919, 0.836727, 0.839393, 0.788605]	262	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	175	9.850409e−03	[0.798542, 0.854101, 0.854315, 0.806106]	5	
$i=1$	177	9.779958e−03	[0.806808, 0.861036, 0.861462, 0.808340]	5	0.313
$i=2$	177	1.138628e−02	[0.782243, 0.836331, 0.834043, 0.784920]	5	
$i=3$	176	1.259544e−02	[0.777628, 0.834029, 0.836690, 0.785366]	5	

Table 4
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 2 for computing the best-fitting weights of the rational cubic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	88	7.501697e−03	[0.866362, 0.939911]	7	
$i=1$	88	8.179204e−03	[0.937168, 0.854591]	7	
$i=2$	89	8.617295e−03	[0.851643, 0.936051]	7	
$i=3$	89	6.994487e−03	[0.941680, 0.870211]	7	0.328
$i=4$	89	7.712267e−03	[0.868871, 0.939994]	7	
$i=5$	89	8.573518e−03	[0.935705, 0.849667]	7	
$i=6$	88	8.265148e−03	[0.854832, 0.936924]	7	
$i=7$	89	7.561317e−03	[0.939118, 0.862619]	7	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	88	7.501697e−03	[0.866362, 0.939911]	4	
$i=1$	88	8.179204e−03	[0.937168, 0.854591]	5	
$i=2$	89	8.617295e−03	[0.851643, 0.936051]	4	
$i=3$	89	6.994487e−03	[0.941680, 0.870211]	4	0.187
$i=4$	89	7.712267e−03	[0.868871, 0.939994]	4	
$i=5$	89	8.573518e−03	[0.935705, 0.849667]	4	
$i=6$	88	8.265148e−03	[0.854832, 0.936924]	5	
$i=7$	89	7.561317e−03	[0.939118, 0.862619]	4	

Table 5
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 3 for computing the best-fitting weights of the rational quintic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	130	3.321477e−03	[1.139599, 1.106147, 1.018917, 0.926744]	376	
$i=1$	128	3.823595e−03	[0.935399, 1.025981, 1.117559, 1.148720]	208	2.437
$i=2$	130	3.432370e−03	[1.145902, 1.112684, 1.022914, 0.930366]	274	
$i=3$	129	3.015703e−03	[0.888400, 0.974535, 1.042578, 1.087367]	478	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	130	1.739287e-03	[1.033776, 0.978617, 0.933440, 0.843413]	6	
$i=1$	128	1.726771e-03	[0.838807, 0.927343, 0.970547, 1.027221]	6	0.265
$i=2$	130	1.607762e-03	[1.032087, 0.975694, 0.931319, 0.841457]	6	
$i=3$	129	1.904271e-03	[0.828015, 0.912513, 0.950083, 1.010514]	6	

Table 6
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 3 for computing the best-fitting weights of the rational cubic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	66	3.455816e−03	[0.973239, 0.954270]	7	
$i=1$	65	2.200337e−03	[0.945824, 0.891170]	7	
$i=2$	65	1.947531e−03	[0.895264, 0.946503]	7	
$i=3$	64	4.129708e−03	[0.952065, 0.973126]	7	0.296
$i=4$	66	4.321906e−03	[0.971470, 0.948619]	7	
$i=5$	65	2.062406e−03	[0.947309, 0.899389]	7	
$i=6$	65	1.708276e−03	[0.916043, 0.951027]	7	
$i=7$	65	6.755706e−03	[0.936736, 0.969365]	7	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	66	3.455816e−03	[0.973239, 0.954270]	4	
$i=1$	65	2.200337e−03	[0.945824, 0.891170]	4	
$i=2$	65	1.947531e−03	[0.895264, 0.946503]	4	
$i=3$	64	4.129708e−03	[0.952065, 0.973126]	4	0.141
$i=4$	66	4.321906e−03	[0.971470, 0.948619]	4	
$i=5$	65	2.062406e−03	[0.947309, 0.899389]	4	
$i=6$	65	1.708276e−03	[0.916043, 0.951027]	4	
$i=7$	65	6.755706e−03	[0.936736, 0.969365]	4	

Table 7
Comparison of error measures (E^i_{RMS}), optimal solutions ($\mathbf{w}^{i,*}$), numbers of iterations (Nit) and computing time (Time) for `lsqcurvefit` and Algorithm 1 when they are tested on data set 4 for computing the best-fitting weights of the rational quintic model

`lsqcurvefit`					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	74	1.278429e−03	[1.069575, 0.972395, 0.930822, 1.021725]	22	
$i=1$	75	1.030284e−02	[1.007062, 2.307917, 2.187811, 0.817368]	37	
$i=2$	76	3.608295e−03	[1.305328, 1.076340, 0.799544, 0.691877]	169	
$i=3$	74	2.771231e−03	[0.559891, 0.675602, 0.606955, 0.595069]	115	1.219
$i=4$	75	1.421212e−03	[0.994152, 0.877017, 0.831445, 0.938514]	79	
$i=5$	74	8.635112e−03	[0.843162, 1.832346, 1.696077, 0.683963]	49	
$i=6$	75	3.532453e−03	[1.825726, 1.359416, 0.955129, 0.874475]	22	
$i=7$	74	2.946511e−03	[0.576109, 0.679351, 0.591689, 0.602577]	133	
Algorithm 1					
Curve segment	M_i	E^i_{RMS}	$\mathbf{w}^{i,*}$	Nit	Time (s)
$i=0$	74	1.102737e-03	[0.984324, 0.923414, 0.849825, 0.962449]	5	
$i=1$	75	9.922827e-03	[3.244976, 7.916870, 7.953229, 2.987525]	13	
$i=2$	76	3.262948e-03	[0.990607, 0.880050, 0.701977, 0.556298]	6	
$i=3$	74	2.769366e-03	[0.556354, 0.674249, 0.603909, 0.595455]	6	0.343
$i=4$	75	1.297896e-03	[0.944059, 0.855645, 0.784090, 0.900877]	6	
$i=5$	74	7.785963e-03	[0.505213, 0.638850, 0.448414, 0.224541]	6	
$i=6$	75	3.006249e-03	[1.095881, 0.903639, 0.716018, 0.586672]	7	
$i=7$	74	2.926852e-03	[0.562427, 0.673401, 0.581701, 0.603127]	6	

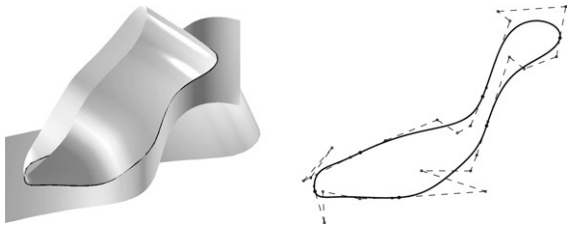

Fig. 1. Least-squares fitting of data set 1 through 6 rational quintic Hermite elements. In both pictures the piecewise quintic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,2301}$ lying on the free-form surfaces intersection is denoted by a solid line. Control polygons of the 6 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,6}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_6$) are shown as dashed lines.

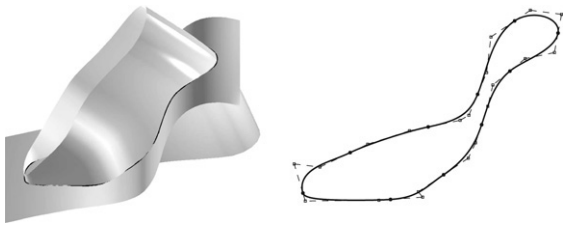

Fig. 2. Least-squares fitting of data set 1 through 10 rational cubic Hermite elements. In both pictures the piecewise cubic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,2301}$ lying on the free-form surfaces intersection is denoted by a solid line. Control polygons of the 10 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,10}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_{10}$) are shown as dashed lines.

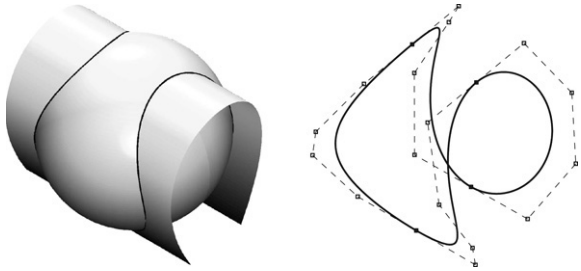

Fig. 3. Least-squares fitting of data set 2 through 4 rational quintic Hermite elements. In both pictures the piecewise quintic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,701}$ lying on the sphere/free-form surface intersection is denoted by a solid line. Control polygons of the 4 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,4}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_4$) are shown as dashed lines.

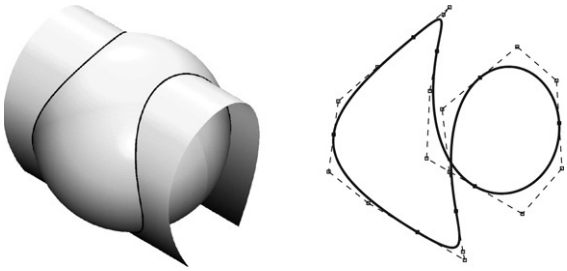

Fig. 4. Least-squares fitting of data set 2 through 8 rational cubic Hermite elements. In both pictures the piecewise cubic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,701}$ lying on the sphere/free-form surface intersection is denoted by a solid line. Control polygons of the 8 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,8}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_8$) are shown as dashed lines.

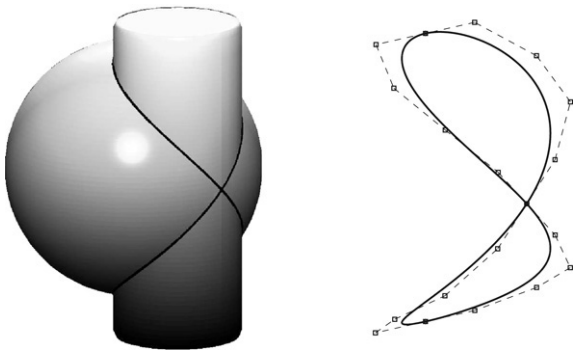

Fig. 5. Least-squares fitting of data set 3 through 4 rational quintic Hermite elements. In both pictures the piecewise quintic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,513}$ lying on the sphere/cylinder intersection is denoted by a solid line. Control polygons of the 4 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,4}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_2 \equiv \mathbf{F}_4$) are shown as dashed lines.

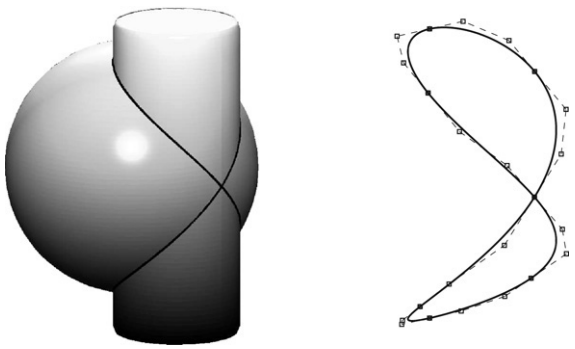

Fig. 6. Least-squares fitting of data set 3 through 8 rational cubic Hermite elements. In both pictures the piecewise cubic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\ldots,513}$ lying on the sphere/cylinder intersection is denoted by a solid line. Control polygons of the 8 curve pieces passing through the assigned break points $\{\mathbf{F}_i\}_{i=0,\ldots,8}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_4 \equiv \mathbf{F}_8$) are shown as dashed lines.

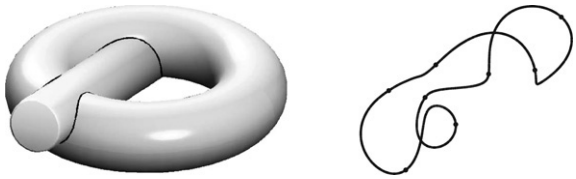

Fig. 7. Least-squares fitting of data set 4 through 8 rational quintic Hermite elements. In both pictures the piecewise quintic that best approximates the sequence of points $\{\mathbf{Q}_k\}_{k=0,\dots,589}$ lying on the torus/cylinder intersection is denoted by a solid line. The assigned break points $\{\mathbf{F}_i\}_{i=0,\dots,8}$ (with $\mathbf{F}_0 \equiv \mathbf{F}_8$) interpolated by the resulting piecewise quintic are denoted by small circles.

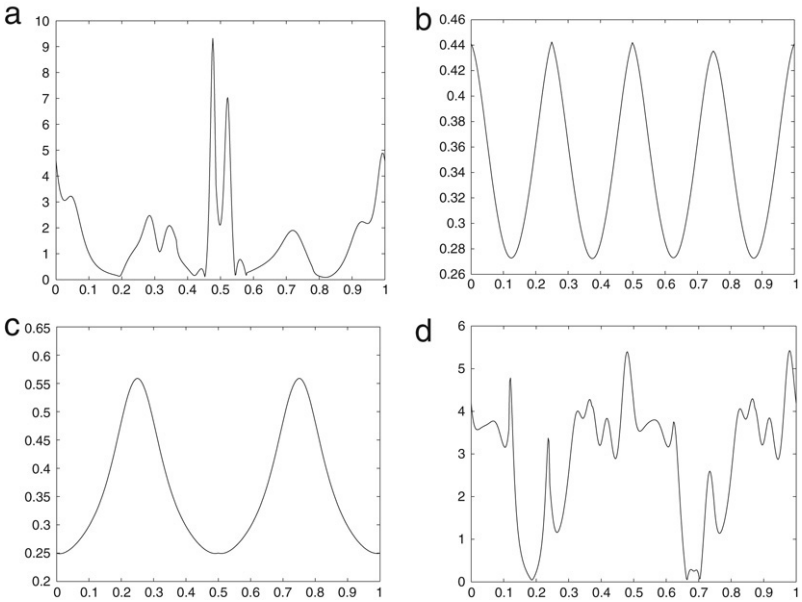

Fig. 8. Curvature plot of the piecewise quintic fitting curve in: (a) Fig. 1, (b) Fig. 3, (c) Fig. 5 and (d) Fig. 7.

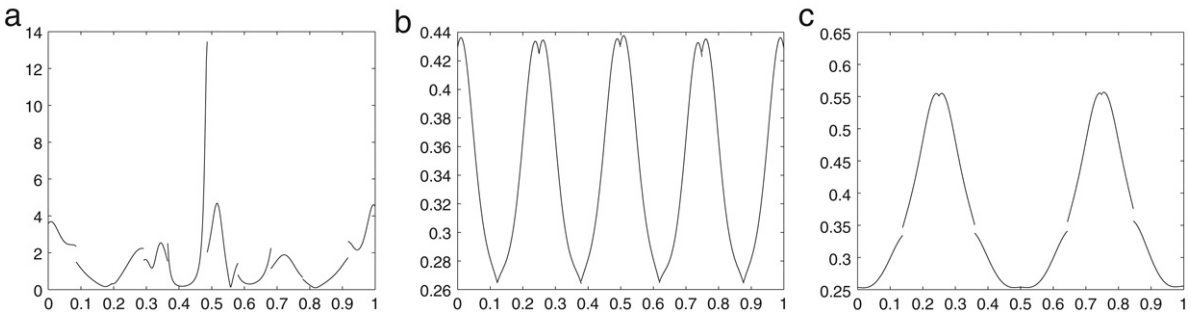

Fig. 9. Curvature plot of the piecewise cubic fitting curve in: (a) Fig. 2, (b) Fig. 4 and (c) Fig. 6.

5. Concluding remarks

Existing techniques for computing a smooth parametric curve that approximates a well-ordered sequence of distinct data satisfying specific requirements on points and derivatives to be interpolated, rely on mathematical models defined in the space of conventional polynomial splines and NURBS. While alternative approaches have proposed using piecewise Hermite polynomials of degree three and five as possible fitting curve basis [7], their rational counterparts have never been taken into account. This consideration prompted us to propose an innovative solution to the problem

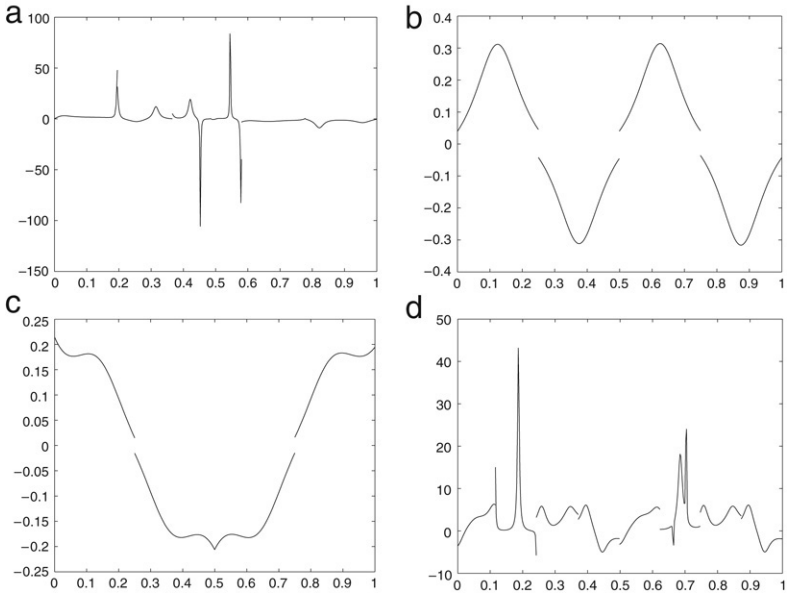

Fig. 10. Torsion plot of the piecewise quintic fitting curve in: (a) Fig. 1, (b) Fig. 3, (c) Fig. 5 and (d) Fig. 7.

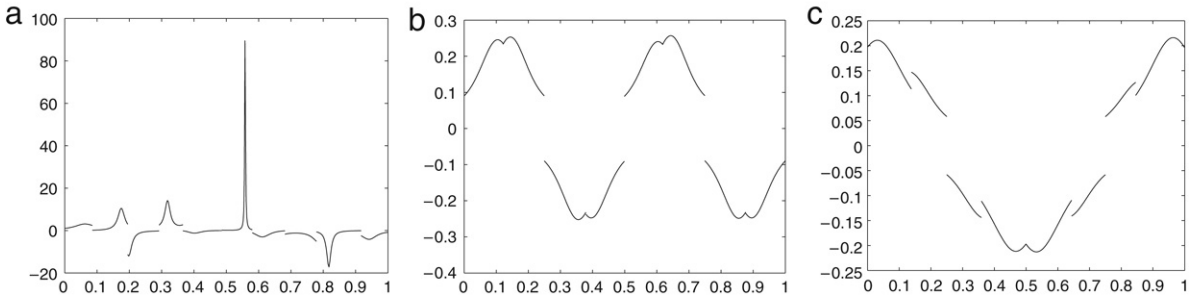

Fig. 11. Torsion plot of the piecewise cubic fitting curve in: (a) Fig. 2, (b) Fig. 4 and (c) Fig. 6.

of constrained least-squares data-fitting that is based on cubic and quintic piecewise rational Hermite interpolants. The use of this novel curve primitive provides an efficient fitter, which turns out to possess a number of advantages that should not be ignored:

- it is simple in definition and construction;
- it can be written into standard NURBS format and thus can be exported to any CAD system;
- it is fully automatic since no user intervention is ever required;
- it provides fair (and, whenever $n > 3$, curvature continuous) reconstructions;
- it is both flexible and powerful as it can easily solve any fitting problem with very few computations;
- it converges rapidly (quadratically) to the target shape, always guaranteeing a highly accurate fitting with very low number of iterations.

All these benefits are due to the fact that the novel fitting procedure combines a very effective optimization algorithm for identifying the best-fitting or optimal weights, with a very powerful rational representation that is able to realize the full modelling potential of NURBS in a cheaper way than conventional approaches because all the control variables are handled collectively and simultaneously in a unified way. Since the NURBS control points turn out to be automatically defined through the weights, the optimization process can be performed only on the latter, so reducing difficulties and computations. Furthermore, due to the locality of the proposed strategy, processing extremely large data sets is not a burden and, if the number of given points is increased, neither instability nor memory problems occur.

The proposed technique results therefore in an efficient and practical method to be used directly in applications such as the reconstruction of a surface/surface intersection curve and the generation of the curve network at the base of a Gordon-type surface, where a fair shape is desired by approximating a given point set with positional constraints.

Acknowledgements

This research was supported by University of Bologna and University of Milano-Bicocca, Italy.

References

[1] G. Casciola, L. Romani, Rational interpolants with tension parameters, in: T. Lyche, M.-L. Mazure, L.L. Schumaker (Eds.), Curve and Surface Design: Saint-Malo 2002, Nashboro Press, 2003, pp. 41–50.
[2] G. Casciola, L. Romani, A piecewise rational quintic Hermite interpolant for use in CAGD, in: M. Dæhlen, K. Mørken, L.L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces: Tromsø2004, Nashboro Press, 2005, pp. 39–49.
[3] J.C. Chambelland, M. Daniel, J.M. Brun, An iterative method for rational pole curve fitting, in: WSCG'2006 Short Paper Conference Proceedings, 2006, pp. 39–46.
[4] J.J. Chou, L.A. Piegl, Data reduction using cubic rational B-splines, IEEE Computer Graphics & Applications 12 (3) (1992) 60–68.
[5] T.F. Coleman, Y. Li, On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds, Mathematical Programming 67 (2) (1994) 189–224.
[6] T.F. Coleman, Y. Li, An interior, trust region approach for nonlinear minimization subject to bounds, SIAM Journal on Optimization 6 (1996) 418–445.
[7] L. Fang, D.C. Gossard, Multidimensional curve fitting to unorganised data points by nonlinear minimization, Computer Aided Design 27 (1) (1995) 48–58.
[8] M. Heinkenschloss, M. Ulbrich, S. Ulbrich, Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption, Mathematical Programming A86 (1999) 615–635.
[9] J. Hoschek, F.J. Schneider, Approximate conversion and data compression of integral and rational B-spline surfaces, in: P.J. Laurent, A. Le Méhauté, L.L. Schumaker (Eds.), Curves and Surfaces in Geometric Design, A.K. Peters Ltd., 1994, pp. 241–250.
[10] C. Kanzow, A. Klug, On affine-scaling interior-point Newton-methods for nonlinear minimization with bound constraints, Computational Optimization and Applications 35 (2006) 177–197.
[11] P. Laurent-Gengoux, M. Mekhilef, Optimization of a NURBS representation, Computer Aided Design 25 (11) (1993) 699–710.
[12] W. Ma, J.-P. Kruth, NURBS curve and surface fitting and interpolation, in: M. Dæhlen, T. Lyche, L.L. Schumaker (Eds.), Mathematical Methods for Curves and Surfaces, Vanderbilt University Press, 1995, pp. 315–322.
[13] W. Ma, J.-P. Kruth, NURBS curve and surface fitting for reverse engineering, International Journal of Advanced Manufacturing Technology 14 (1998) 918–927.
[14] M. Sarfraz, Fitting curve to planar digital data, in: Proceedings of the Sixth International Conference on Information Visualization (IV'02), 2002, pp. 633–638.
[15] M. Sarfraz, Optimal curve fitting to digital data, Journal of the WSCG 11 (1) (2003).